10-1-2007

Estimating Quality of Support Vector Machines Learning Under Probabilistic and Interval Uncertainty: Algorithms and Computational Complexity

Canh Hao Nguyen

Tu Bao Ho

Vladik Kreinovich
University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-07-54
Published in: Van-Nam Huynh, Yoshiteru Nakamori, Hiroakira Ono, Jonathan Lawry, Vladik Kreinovich, and Hung T. Nguyen (eds.), *Interval/Probabilistic Uncertainty and Non-Classical Logics*, Springer-Verlag, Berlin-Heidelberg-New York, 2008, pp. 57-69.

Estimating Quality of Support Vector Machines Learning Under Probabilistic and Interval Uncertainty: Algorithms and Computational Complexity

Canh Hao Nguyen[1], Tu Bao Ho[1], and Vladik Kreinovich[2]

[1] School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan, {canhhao,bao}@jaist.ac.jp

[2] Department of Computer Science, University of Texas, El Paso, TX 79968-0518, USA, vladik@utep.edu

Summary. Support Vector Machines (SVM) is one of the most widely used technique in machines leaning. After the SVM algorithms process the data and produce some classification, it is desirable to learn how well this classification fits the data. There exist several measures of fit, among them the most widely used is kernel target alignment. These measures, however, assume that the data are known exactly. In reality, whether the data points come from measurements or from expert estimates, they are only known with uncertainty. As a result, even if we know that the classification perfectly fits the nominal data, this same classification can be a bad fit for the actual values (which are somewhat different from the nominal ones). In this paper, we show how to take this uncertainty into account when estimating the quality of the resulting classification.

1 Formulation of the Problem

Machine learning: main problem. In many practical situations, we have examples of several types of objects, and we would like to use these examples to teach the computers to distinguish between objects of different types. Each object can be characterized by the corresponding values of several relevant quantities. If we denote the number of these quantities by d, then we can say that each object i can be represented by a d-dimensional vector $x^{(i)} = (x_1^{(i)}, \ldots, x_k^{(i)}, \ldots, x_d^{(i)})$, where $x_k^{(i)}$ denotes the value of the k-th quantity for i-th object. So, from the mathematical viewpoint, the problem is as follows: in d-dimensional space X, we have several points $x^{(1)}, \ldots, x^{(n)}$ belonging to different classes, and we need to be able, given a new point $x \in X$, to assign it to one of these classes.

In the simplest case when we have two classes, we have several points belonging to the first class, and several points which do not belong to the first class, and we must find a separating algorithm.

Linear classification: main idea. In the past, a typical approach to data classification was to find a hyperplane $c_1 \cdot x_1 + \ldots + c_d \cdot x_d = c_0$ which separates the two classes – so that $c_1 \cdot x_1^{(i)} + \ldots + c_d \cdot x_d^{(i)} > c_0$ for all *positive* examples (i.e., examples from the first class) and $c_1 \cdot x_1^{(i)} + \ldots + c_d \cdot x_d^{(i)} < c_0$ for all *negative* examples (i.e., examples which do not belong to the first class).

Linear classification: limitations. The main limitations of linear classification approach is that in many important practical cases, there is no hyperplane separating positive and negative examples.

For example, suppose that we want to teach the computer to distinguish between the center of the city and its suburbs. To do that, we mark several places in the center as positive examples and places in the suburbs as negative examples. Here, a natural idea is to take $d = 2$, so that x_1 and x_2 are two coordinates of each point. To make it easier, we can take the central square of the city as the origin of the coordinate system, i.e., as a point $(0, 0)$.

In this example, separation is straightforward: points whose distance $\sqrt{x_1^2 + x_2^2}$ to the center is below a certain threshold t are within the city center, while points for which the distance is $> t$ are in the suburbs. However, no straight line can separate close points from the distant ones – because on each side of the straight line we have points which are far away from the center.

Support Vector Machines: main idea. What can we do when there is no linear separation? In the 2-D case, as long as there is a separation, i.e., as long as the same point $x \in X$ does not appear as both a positive and a negative example, we can draw a curve separating positive points from negative ones. Similarly, in the d-dimensional case, we can always draw a $(d-1)$-dimensional surface separating positive and negative examples. Moreover, we can always find a continuous function $f(x_1, \ldots, x_d)$ such that $f(x_1^{(i)}, \ldots, x_d^{(i)}) > 0$ for all positive examples and $f(x_1^{(i)}, \ldots, x_d^{(i)}) < 0$ for all negative examples.

A continuous function $f(x_1, \ldots, x_d)$ can be, with arbitrary accuracy, approximated by polynomials; thus, be selecting a good enough accuracy, we can have a polynomial

$$\widetilde{f}(x_1, \ldots, x_d) = c_0 + c_1 \cdot x_1 + \ldots + c_d \cdot x_d + \sum_{k=1}^{d} \sum_{l=1}^{d} c_{kl} \cdot x_k \cdot x_l + \ldots$$

which has the same separating property, i.e.,

$$c_0 + c_1 \cdot x_1^{(i)} + \ldots + c_d \cdot x_d^{(i)} + \sum_{k=1}^{d} \sum_{l=1}^{d} c_{kl} \cdot x_k^{(i)} \cdot x_l^{(i)} + \ldots > 0$$

for all positive examples and

$$c_0 + c_1 \cdot x_1^{(i)} + \ldots + c_d \cdot x_d^{(i)} + \sum_{k=1}^{d} \sum_{l=1}^{d} c_{kl} \cdot x_k^{(i)} \cdot x_l^{(i)} + \ldots < 0$$

for all negative examples. These formulas clearly show that this non-linear separation means that we linearly separate points $(x_1, \ldots, x_n, x_1^2, x_1 \cdot x_2, \ldots)$.

Instead of polynomials, we could use trigonometric polynomials or sums of Gaussian functions, or any other class of approximating functions. In all these cases, what we are doing is mapping each point x into a point $\phi(x) = (\phi_1(x), \ldots, \phi_p(x), \ldots, \phi_N(x))$ in a higher-dimensional space (of dimension $N \geq d$), and then use linear separation to separate the resulting points $\phi(x^{(1)}), \ldots, \phi(x^{(n)})$ in the N-dimensional space. This, in a nutshell, is the main idea behind the Support Vector Machines (SVM) techniques; see, e.g., [10].

Need to estimate classification quality. The fact that we have a surface separating positive examples from negative examples does not necessarily mean that this classification is good. Intuitively, if we have a new example x which is similar to one of the previously given examples $x^{(i)}$, then this new example should be classified to the same class as $x^{(i)}$. So, we want to make sure not only that all the positive examples are on the right side of the separating surface, but also that the points which are close to these examples are also on the same side of the separating surface. In other words, we want to make sure that all the examples are sufficiently far away from the separating surface. Thus, some reasonable measure of the distance from this surface can serve as the measure of the quality of the resulting classification.

Several such criteria have been proposed. These criteria are usually defined in terms of the *kernel matrix* $k_{ij} \stackrel{\text{def}}{=} \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$, where

$$\langle \phi, \phi' \rangle \stackrel{\text{def}}{=} \sum_{p=1}^{N} \phi_p \cdot \phi'_p.$$

KTA. The most widely used criterion is the kernel target alignment (KTA) A [1], which is defined as follows (in our notations):

$$A = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij} \cdot y_i \cdot y_j}{n \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij}^2},$$

where $y_i = 1$ for positive examples and $y_i = -1$ for negative examples. This criterion has a very intuitive meaning. In the ideal situation, the separation should be as sharp as possible: we should have all the vectors $\phi(x^{(i)})$ corresponding to the positive examples to be equal to some unit vector e and all

the vectors corresponding to the negative examples to be equal to $-e$. In this ideal situation, the kernel matrix is equal to $y_i \cdot y_j$. To estimate the quality of a classification, it is reasonable to check how close the actual kernel matrix is to this ideal one. One way to check is to consider both matrices as vectors in a $N \times N$ dimensional space, and estimate the cosine of the angle between these vectors; if the vectors coincide, the angle is 0, and the cosine is 1; if the vectors are close, the angle is close to 0, and the cosine is close to 1, etc. This cosine is equal to

$$\frac{\langle K, \cdot y^T \rangle_F}{\sqrt{\langle K, K \rangle_F \langle y \cdot y^T, y \cdot y^T \rangle_F}},$$

where

$$\langle K, K' \rangle_F \stackrel{\text{def}}{=} \sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij} \cdot k'_{ij},$$

so we get the above expression for the KTA.

Class Separability Measure (CSM). An alternative measure to KTA has been proposed in [13]. This measure is actually defined for a general case of classifying the data into several ($C \geq 2$) classes. The main idea is that in a good classification, data points within each class should be close to each other, while data points from different classes should be far away from each other. In other words, "within-class" scatter should be much smaller than the "between-classes" scatter.

Each class is naturally characterized by its average. Thus, for each data point, its contribution to the "within-class" scatter can be described as a (squared) distance from this data point to the average, and its contribution to the "between-classes" scatter can be described as a (squared) distance between the average of this class and the overall average.

In the SVM approach, each data point $x^{(i)}$ is represented by the vector $\phi(x^{(i)})$. Thus, the above idea can be reformulated as follows. For each class S_c, $c = 1, 2, \ldots, C$, let n_c denote the number of data points classified into this class. Let ϕ_c denote the average of all the vectors $\phi(x^{(i)})$ from the c-th class, and let ϕ denote the average of all n vectors $\phi(x^{(i)})$. Then, we can define the *within-class scatter* s_w as

$$s_w \stackrel{\text{def}}{=} \sum_{c=1}^{C} \sum_{i \in S_c} \|\phi(x^{(i)}) - \phi_c\|^2,$$

and the *between-classes scatter* as

$$s_b \stackrel{\text{def}}{=} \sum_{c=1}^{C} n_c \cdot \|\phi_c - \phi\|^2.$$

We can also define *total scatter* as the sum $s_t \stackrel{\text{def}}{=} s_w + s_b$. A classification is of good quality if $s_w \ll s_b$, i.e., equivalently, if $s_b \approx s_t$ and the ratio $C \stackrel{\text{def}}{=} \dfrac{s_b}{s_t}$ is close to 1. This ratio C is used as an alternative quality characteristic.

For the case of two classes, we will denote the number of the corresponding examples as n^+ and n^-, and the averages of the corresponding vectors $\phi(x^{(i)})$ by ϕ^+ and ϕ^-. The value of the CSM ratio C can be computed in terms of the kernel matrix $k_{ij} = \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$ as follows:

- First, for every i, we compute

$$a_i^+ = \frac{1}{n^+} \sum_{j:y_j=1} k_{ij}; \quad a_i^- = \frac{1}{n^-} \sum_{j:y_j=-1} k_{ij}.$$

- Second, we compute

$$a^{++} = \frac{1}{n^+} \sum_{j:y_j=1} a_i^+, \quad a^{+-} = \frac{1}{n^-} \sum_{j:y_j=-1} a_i^+,$$

$$a^{-+} = \frac{1}{n^+} \sum_{j:y_j=1} a_i^-, \quad a^{--} = \frac{1}{n^+} \sum_{j:y_j=-1} a_i^-,$$

  and $s_b = a^{++} - a^{+-} - a^{-+} + a^{--}$.
- Then, we compute

$$s_w = \sum_{i=1}^{n} k_{ii} - n^+ \cdot a^{++} - n^- \cdot a^{--},$$

  and $C = \dfrac{s_b}{s_b + s_w}$.

A new alternative quality measure: FSM. In many practical examples, KTA and CSM provides a reasonable estimate for the quality of fit, in the sense that cases when we have a better fit have a larger value of KTA or CSM. However, there are examples when the values of KTA and CSM are larger for the cases when intuitively, the classification quality is worse.

One reason for the sometimes counterintuitive character of CSM is that CSM estimates a within-class scatter based on deviations in all directions. For example, if for some coordinate $\phi_p(x)$, we have $\phi_p(x^{(i)}) = 1$ for all positive examples and $\phi_p(x^{(i)}) = -1$ for all negative examples, then intuitively, we have a perfect classification. However, since the values $\phi_q(x^{(i)})$ for $q \neq p$ may be widely scattered, we can have a huge value of the within-class scatter, and thus, a very low value of the CSM measure of fit.

To avoid this problem, it is reasonable to take into account only the scatter in the direction between the centers ϕ^- and ϕ^+. The corresponding Feature-Spaced Measure (FSM) was proposed in [7].

To estimate this measure, we do the following:

- First, we compute the average ϕ^+ of the values $\phi(x^{(i)})$ for all the positive examples and the average ϕ^- of the values $\phi(x^{(i)})$ for all the negative examples. In the ideal case, as we have mentioned, we should have $\phi^+ = e$ and $\phi^- = -e$ for some unit vector e.

- Then, we estimate the vector e as the unit vector in the direction of the difference $\phi^+ - \phi^-$, i.e., as $e = \dfrac{\phi^+ - \phi^-}{\|\phi^+ - \phi^-\|}$.
- Next, for each example i, we compute the projection $p_i = \langle \phi(x^{(i)}), e \rangle$ of the vector $\phi^{(i)}$ to the direction e.
- Finally, we compute the population means

$$p^+ = \frac{1}{n^+} \cdot \sum_{i:y_i=1} p_i; \quad p^- = \frac{1}{n^-} \cdot \sum_{i:y_i=-1} p_i,$$

where n^+ and n^- $(= n - n^+)$ denote the numbers of positive and negative examples, compute population variances

$$V^+ = \frac{1}{n^+ - 1} \cdot \sum_{i:y_i=1} (p_i - p^+)^2; \quad V^- = \frac{1}{n^- - 1} \cdot \sum_{i:y_i=-1} (p_i - p^-)^2,$$

and the desired value

$$\frac{\sqrt{V^+} + \sqrt{V^-}}{\|\phi^+ - \phi^-\|}.$$

This algorithm describes how to compute these values based on the vectors $\phi(x^{(i)})$; alternatively, as shown in [7], we can compute it from the kernel matrix k_{ij} as follows:

- First, we compute the values a_i^+ and a_i^- as in the CSM case.
- Second, we compute the values a^{++}, a^{+-}, a^{-+}, and a^{--} as in the CSM case, and compute $\|\phi^+ - \phi^-\|^2 = a^{++} - a^{+-} - a^{-+} + a^{--}$.
- Then, we compute

$$V^+ = \frac{\sum\limits_{i:y_i=1} ((a_i^- - a^{-+}) - (a_i^+ - a^{++}))^2}{(n^+ - 1) \cdot \|\phi^+ - \phi^-\|^2},$$

$$V^- = \frac{\sum\limits_{i:y_i=-1} ((a_i^+ - a^{+-}) - (a_i^- - a^{--}))^2}{(n^- - 1) \cdot \|\phi^+ - \phi^-\|^2},$$

and the desired value

$$\frac{\sqrt{V^+} + \sqrt{V^-}}{\|\phi^+ - \phi^-\|}.$$

2 How to Take into Account Probabilistic and Interval Uncertainty: Formulation of the Problem and Linearized Algorithms for Solving this Problem

Need to take into account probabilistic and interval uncertainty. In presenting algorithms for computing the SVM quality measures, we (implicitly) assumed

that we know the exact values of the data points $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})$. In reality, the values $x_k^{(i)}$ come from measurements or from expert estimates, and both measurements and expert estimates are never 100% accurate. As a result, the measured (estimated) values $\widetilde{x}_k^{(i)}$ of the corresponding quantities are, in general, different from the (unknown) actual values $x_k^{(i)}$.

It is desirable to take into account this measurement (estimation) uncertainty when estimating the quality measures.

Need to describe measurement and/or estimation uncertainty. In order to gauge the effect of the measurement (estimation) uncertainty on the values of the quality measures, we must have the information about the measurement (estimation) uncertainty, i.e., the information about the measurement (estimation) errors $\Delta x_k^{(i)} \stackrel{\text{def}}{=} \widetilde{x}_k^{(i)} - x_k^{(i)}$.

For simplicity, in the following text, we will mainly talk about measurement errors; for estimation errors the situation is very similar.

Upper bound on the measurement error. How can the measurement error be described? First, the manufacturer of a measuring instrument must provide us with an upper bound Δ on the absolute value $|\Delta x|$ of the measurement error Δx. If no such bound was guaranteed, this would mean that the difference Δx can be arbitrarily large; in this situation, after getting a measurement result say $\widetilde{x} = 1$, we cannot be sure whether the actual value x of the measured quantity is 1, 0, 10, 100, or 1,000,000. In this situation, $\widetilde{x} = 1$ is a wild guess, not a measurement result.

When we know this upper bound Δ, this means that the actual value Δx of the measurement error must be inside the interval $[-\Delta, \Delta]$.

Probabilistic information. In addition to the upper bound Δ, we often also know the probabilities of different values Δx from the interval $[-\Delta, \Delta]$.

This situation of probabilistic uncertainty is traditionally used in engineering and scientific practice. Most frequently, scientists and engineers consider the situation when the measurement error is normally distributed, with 0 mans and known standard deviation σ; see, e.g,., [9].

Case of interval uncertainty. In many important practical situations, we do not have the information about the probabilities of different values of Δx, we only know the upper bound Δ.

The reason is that the probabilistic information usually comes from comparing the results of measuring the same quantity with two different measuring instruments: the one used for actual measurements and the *standard* (much more accurate) one – whose results are so much closer to the actual values that we can ignore the corresponding measurement errors and consider these results actual values.

There are two situations when this comparison is not done. The first such situation is the situation of cutting-edge measurements, when we are actually using the best possible measuring instrument. For example, if we perform

some protein measurements by using a state-of-the-art electronic microscope, it would be nice to be able to compare the results with a much more accurate microscope – but ours is already the best.

Another case when the probabilities are not determined is when we have limited resources. For example, in geophysics, in every seismic experiment, we use a large number of sensors to measure the corresponding time delays. It would be nice to be able to compare all these sensors with more accurate ones. However, the detailed comparison of each sensor requires the use of costly standard sensors and, as a result, costs several orders of magnitude more than the cost of buying a new sensor – so we often cannot do this detailed probabilistic "calibration" within our limited resources.

In both cases, the only information we have about the measurement error $\Delta x = \widetilde{x} - x$ is the upper bound Δ: $|\Delta x| \le \Delta$. In such situations, once we have the measurement result $\widetilde{x}$, the only conclusion that we can make about the (unknown) actual value x is that x belongs to the interval $\mathbf{x} = [\underline{x}, \overline{x}]$, where $\underline{x} \stackrel{\text{def}}{=} \widetilde{x} - \Delta$ and $\overline{x} \stackrel{\text{def}}{=} \widetilde{x} + \Delta$. This situation is called the situation of *interval uncertainty.*

Estimating the measures of fit under measurement uncertainty: formulation of the problem. In general, we have an algorithm

$$Q(x_1^{(1)}, \ldots, x_d^{(1)}, x_1^{(2)}, \ldots, x_d^{(2)}, \ldots, x_1^{(n)}, \ldots, x_d^{(n)})$$

which transforms the values

$$x_1^{(1)}, \ldots, x_d^{(1)}, x_1^{(2)}, \ldots, x_d^{(2)}, \ldots, x_1^{(n)}, \ldots, x_d^{(n)}$$

of the corresponding quantities into the value

$$y = Q(x_1^{(1)}, \ldots, x_d^{(1)}, x_1^{(2)}, \ldots, x_d^{(2)}, \ldots, x_1^{(n)}, \ldots, x_d^{(n)})$$

of the corresponding quality characteristic. Due to measurement errors, we do not know the actual values $x_k^{(i)}$. Instead, we only know the intervals $[\underline{x}_k^{(i)}, \overline{x}_k^{(i)}]$ of possible values of $x_k^{(i)}$ – and possible also the probabilities of different values from these intervals.

Different values $x_k^{(i)} \in [\underline{x}_k^{(i)}, \overline{x}_k^{(i)}]$ lead, in general, to different values of the measure of fit $y = Q(x_1^{(1)}, \ldots, x_d^{(n)})$. It is therefore desirable to find the range $\mathbf{y}$ of possible values of y:

$$\mathbf{y} = \{Q(x_1^{(1)}, \ldots, x_d^{(n)}) \,|\, x_1^{(1)} \in [\underline{x}_1^{(1)}, \overline{x}_1^{(1)}], \ldots, x_d^{(n)} \in [\underline{x}_d^{(n)}, \overline{x}_d^{(n)}]\},$$

and, if possible, the probability of different values of y within this interval.

Case of relatively small measurement error: possibility of linearization. When the measurement errors Δx_i are relatively small, we can use linearization.

By definition of the measurement error $\Delta x_k^{(i)} = \widetilde{x}_k^{(i)} - x_k^{(i)}$, hence $x_k^{(i)} = \widetilde{x}_k^{(i)} - \Delta x_k^{(i)}$. When the measurement errors $\Delta x_k^{(i)}$ of direct measurements are relatively small, we can expand the expression

$$\Delta y = \widetilde{y} - y = Q(\widetilde{x}_1^{(1)}, \ldots, \widetilde{x}_d^{(n)}) - Q(x_1^{(1)}, \ldots, x_d^{(n)}) =$$

$$Q(\widetilde{x}_1^{(1)}, \ldots, \widetilde{x}_d^{(n)}) - Q(\widetilde{x}_1^{(1)} - \Delta x_1^{(1)}, \ldots, \widetilde{x}_d^{(n)} - \Delta x_d^{(n)})$$

in Taylor series and only keep linear terms in the resulting expansion. Since

$$y = Q(\widetilde{x}_1^{(1)} - \Delta x_1^{(1)}, \ldots, \widetilde{x}_d^{(n)} - \Delta x_d^{(n)}) \approx Q(\widetilde{x}_1^{(1)}, \ldots, \widetilde{x}_d^{(n)}) - \sum_{i=1}^{n} \sum_{k=1}^{d} \frac{\partial Q}{\partial x_k^{(i)}} \cdot \Delta x_k^{(i)},$$

we conclude that $\Delta y = \widetilde{y} - y = \sum\limits_{i=1}^{n} \sum\limits_{k=1}^{d} c_k^{(i)} \cdot \Delta x_k^{(i)}$, where $c_k^{(i)} \stackrel{\text{def}}{=} \dfrac{\partial Q}{\partial x_k^{(i)}}$.

Linearization: probabilistic case. When $\Delta x_k^{(i)}$ are independent normally distributed random variables with 0 means and known standard deviations $\sigma_k^{(i)}$, the linear combination $\Delta y = \sum\limits_{i=1}^{n} \sum\limits_{k=1}^{d} c_k^{(i)} \cdot \Delta x_k^{(i)}$ is also normally distributed, with 0 mean and standard deviation

$$\sigma = \sqrt{\sum_{i=1}^{n} \sum_{k=1}^{d} (c_k^{(i)} \cdot \sigma_k^{(i)})^2}.$$

So, in this case, to find the uncertainty in the value of the measure of fit, it is sufficient to be able to compute the values of the corresponding partial derivatives $c_k^{(i)}$.

Linearization: interval case. The dependence of Δy on $\Delta x_k^{(i)}$ is linear: it is increasing relative to $x_k^{(i)}$ if $c_k^{(i)} \geq 0$ and decreasing if $c_k^{(i)} < 0$. So, to find the largest possible value Δ of Δy, we must take:

- the largest possible value $\Delta x_k^{(i)} = \Delta_k^{(i)}$ when $c_k^{(i)} \geq 0$, and
- the smallest possible value $\Delta x_k^{(i)} = -\Delta_k^{(i)}$ when $c_k^{(i)} < 0$.

In both cases, the corresponding term in the sum has the form $|c_k^{(i)}| \cdot \Delta_k^{(i)}$, so we can conclude that

$$\Delta = \sum_{i=1}^{n} \sum_{k=1}^{d} |c_k^{(i)}| \cdot \Delta_k^{(i)}.$$

Similarly, the smallest possible value of Δy is equal to $-\Delta$. Thus, the range of possible values of y is equal to $[\underline{y}, \overline{y}] = [\widetilde{y} - \Delta, \widetilde{y} + \Delta]$. So, to compute Δ, it is also sufficient to know the partial derivatives $c_k^{(i)}$.

How to compute the derivatives. For all the above characteristics y, we have an explicit expression in terms of the values k_{ij}. Thus, we can find the explicit analytic formulas in terms of the corresponding derivatives as

$$c_k^{(i)} = \sum_{a=1}^{n} \sum_{b=1}^{n} \frac{\partial y}{\partial k_{ab}} \cdot \frac{\partial k_{ab}}{\partial x_k^{(i)}}.$$

Here, the first partial derivative can be explicitly computed: e.g., for KTA $Q = A$, we have

$$\frac{\partial y}{\partial k_{ab}} = \frac{y_a \cdot y_b}{n \cdot \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} k_{ij}^2} - 2k_{ab} \cdot \frac{\sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} k_{ij} \cdot y_i \cdot y_j}{n \cdot \left(\sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} k_{ij}^2 \right)^2}.$$

For $k_{ab} = \sum\limits_{p=1}^{N} \phi_p(x^{(a)}) \cdot \phi_p(x^{(b)})$, the derivative $\dfrac{\partial k_{ab}}{\partial x_k^{(i)}}$ is only different from 0 if $a = i$ or $b = i$:

$$\frac{\partial k_{ib}}{\partial x_k^{(i)}} = \sum_{p=1}^{N} \frac{\partial \phi_p}{\partial x_k}(x^{(i)}) \cdot \phi_p(x^{(b)}) \quad \text{for} \quad a = i \quad \text{and} \quad b \neq i;$$

$$\frac{\partial k_{ai}}{\partial x_k^{(i)}} = \sum_{p=1}^{N} \phi_p(x^{(a)}) \cdot \frac{\partial \phi_p}{\partial x_k}(x^{(i)}) \quad \text{for} \quad a \neq i \quad \text{and} \quad b = i;$$

$$\frac{\partial k_{ii}}{\partial x_k^{(i)}} = 2 \sum_{p=1}^{N} \frac{\partial \phi_p}{\partial x_k}(x^{(i)}) \cdot \phi_p(x^{(i)}) \quad \text{for} \quad a = b = i.$$

3 In General, Estimating Quality of SVM Learning Under Interval Uncertainty Is NP-Hard

Motivations. In the previous section, we considered the case when measurement errors are small, e.g., no more than 10%, so that we can ignore terms which are quadratic in terms of these errors. For example, for $10\% = 0.1$, the quadratic terms are proportional to $0.1^2 = 1\% \ll 10\%$ and thus, indeed, much smaller than the original errors. In this case, we can linearize the formulas for the quality of SVM learning and get efficient algorithms for computing the range of the corresponding quality characteristics.

In practice, however, the measurement errors are often not very small. For example, for a realistic measurement error of 30%, the square is $\approx 10\%$ and is no longer negligible in comparison with the original measurement errors. In such situations, we can no longer use linearized techniques, we must consider the original problem of computing the range $[\underline{y}, \overline{y}]$ of a given characteristic $Q(x_1^{(1)}, \ldots, x_d^{(n)})$ under interval uncertainty:

$$[\underline{y}, \overline{y}] = \{Q(x_1^{(1)}, \ldots, x_d^{(n)}) \,|\, x_1^{(1)} \in [\underline{x}_1^{(1)}, \overline{x}_1^{(1)}], \ldots, x_d^{(n)} \in [\underline{x}_d^{(n)}, \overline{x}_d^{(n)}]\}.$$

It turns out that in general, this problem is NP-hard – at least it is NP-hard for the most widely used measures of fit KTA and CSM.

Crudely speaking, NP-hard means that there is practically no hope of designing an efficient algorithm which would always correct compute this range; for precise definitions, see, e.g., [3, 4, 8].

Theorem 1. *Computing the range of KTA under interval uncertainty is NP-hard.*

Proof. To prove NP-hardness of our problem, we will reduce a known NP-hard problem to our problem of computing the range **A** of KTA A under interval uncertainty. Specifically, we will reduce, to our problem, the following *partition* problem [3] that is known to be NP-hard:

- Given k positive integers $s_1, \ldots, s_k$,
- check whether it is possible to find the values $\varepsilon_i \in \{-1, 1\}$ for which $\sum_{i=1}^{k} \varepsilon_i \cdot s_i = 0$.

To each instance $s_1, \ldots, s_k$ of this problem, we assign the following instance of the problem of computing **A**: we take $d = 1$, $n = k + 1$, $y_1 = \ldots = y_k = 1$, $y_{k+1} = -1$, $\mathbf{x}^{(i)} = [-s_i, s_i]$ for $i \le k$, and $\mathbf{x}^{(n)} = \{2S\}$, where $S \stackrel{\text{def}}{=} \max_{i=1,\ldots,k} s_i$. As ϕ, we take a 2-dimensional mapping $\phi = (\phi_1, \phi_2)$ consisting of the following two piece-wise linear functions:

$$\phi_1(x) = \begin{cases} x & \text{if } x \le S \\ 2S - x & \text{if } S \le x \le 2S \\ 0 & \text{if } x \ge 2S \end{cases}; \quad \phi_2(x) = \begin{cases} 0 & \text{if } x \le S \\ x/S - 1 & \text{if } S \le x \le 2S \\ 1 & \text{if } x \ge 2S \end{cases}.$$

In this case,

$$k_{ij} = \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle = \begin{cases} x^{(i)} \cdot x^{(i)} & \text{if } i, j < n, \\ 1 & \text{if } i = j = n, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij} \cdot y_i \cdot y_j = \sum_{i=1}^{k} \sum_{j=1}^{k} x^{(i)} \cdot x^{(i)} + 1 = \left(\sum_{i=1}^{k} x^{(i)} \right)^2 + 1;$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij}^2 = \sum_{i=1}^{k} \sum_{j=1}^{k} \left(x^{(i)} \right)^2 \cdot \left(x^{(j)} \right)^2 + 1 = \left(\sum_{i=1}^{k} \left(x^{(i)} \right)^2 \right)^2 + 1; \text{ and}$$

$$A = \frac{\left(\sum_{i=1}^{k} x^{(i)}\right)^2 + 1}{n \cdot \sqrt{\left(\sum_{i=1}^{k} \left(x^{(i)}\right)^2\right)^2 + 1}}.$$

The numerator is always greater than or equal to 1. Since $|x^{(i)}| \le s_i$, we have $\left(x^{(i)}\right)^2 \le s_i^2$ and hence, the denominator is always $\le n \cdot \sqrt{\left(\sum_{i=1}^{k} s_i^2\right)^2 + 1}$. Thus, we always have $A \ge A_0 \stackrel{\text{def}}{=} \frac{1}{n \cdot \sqrt{\left(\sum_{i=1}^{k} s_i^2\right)^2 + 1}}$. The only possibility for $A = A_0$ is when the numerator of the fraction A is equal to 1, and its denominator is equal to $n \cdot \sqrt{\left(\sum_{i=1}^{k} s_i^2\right)^2 + 1}$. This is only possible when $|x^{(i)}| = s_i$ for all i, i.e., when $x^{(i)} = \varepsilon_i \cdot s_i$ for some $\varepsilon_i \in \{-1, 1\}$, and $\sum_{i=1}^{k} x^{(i)} = 0$ – i.e., exactly when the original instance of the partition problem has a solution. So, $\underline{A} = A_0$ if and only if the original instance has a solution. This reduction proves that our problem is indeed NP-hard.

Theorem 2. *Computing the range of CSM under interval uncertainty is NP-hard.*

Proof. Under the same reduction as in Theorem 1, we get $n^+ = k$, $n^- = 1$, $a_n^+ = 0$ and for $i < n$, we have $a_i^+ = \frac{1}{k} \cdot \sum_{j=1}^{k} x^{(i)} \cdot x^{(j)} = x^{(i)} \cdot E$, where $E \stackrel{\text{def}}{=} \frac{1}{k} \cdot \sum_{i=1}^{k} x^{(i)}$. Similarly, $a_n^- = 1$ and $a_i^- = 0$ for all $i < n$. Thus, $a^{++} = \frac{1}{k} \cdot \sum_{i=1}^{k} a_i^+ = E \cdot \frac{1}{k} \cdot \sum_{i=1}^{k} x^{(i)} = E^2$, $a^{+-} = a^{-+} = 0$, and $a^{--} = 1$. Hence, $s_b = E^2 + 1$, $s_w = \sum_{i=1}^{k} \left(x^{(i)}\right)^2 - k \cdot E^2 - 1$ and thus, $C = \frac{E^2 + 1}{\sum_{i=1}^{k} \left(x^{(i)}\right)^2 - (k-1) \cdot E^2}$. The numerator is ≥ 1, the denominator is $\le \sum_{i=1}^{k} s_i^2$, hence $C \ge C_0 \stackrel{\text{def}}{=} \frac{1}{\sum_{i=1}^{k} s_i^2}$. The only possibility to have $C = C_0$ is when $E = 0$ and $|x^{(i)}| = s_i$ for all i, i.e., when the original instance of the partition problem has a solution. The theorem is proven.

4 Conclusion

For classification produced by machine learning techniques, it is desirable to learn how well this classification fits the data. There exist several measures of fit, among them the most widely used is kernel target alignment.

The existing formulas for these measures assume that the data are known exactly. In reality, whether the data points come from measurements or from expert estimates, they are only known with uncertainty. As a result, even if we know that the classification perfectly fits the nominal data, this same classification can be a bad fit for the actual values (which are somewhat different from the nominal ones). In this paper, we show how, when the measurement errors are relatively small, we can take this uncertainty into account when estimating the quality of the resulting classification. We also show that in the general case of large uncertainty, the problem of estimating the range of these measures of fit is NP-hard.

Acknowledgments. Vladik Kreinovich supported in part by NSF grants HRD-0734825, EAR-0225670, and EIA-0080940, by Texas Department of Transportation grant No. 0-5453, and by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08.

The authors are thankful to the anonymous referees for valuable comments.

References

1. Cristianini N, Shawe-Taylor J, Elisseeff A, Kandola J (2002) On kernel-target alignment. In: Dietterich TG, Becker S, Ghahramani Z (eds) Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, Massachusetts
2. Ferson S, Ginzburg L, Kreinovich V, Longpré L, Aviles M (2002) Computing variance for interval data is NP-hard. SIGACT News 33(2):108–118
3. Garey MR, Johnson DS (1979) Computers and Intractability, a Guide to the Theory of NP-Completeness. WH Freeman and Company, San Francisco, California
4. Kreinovich V, Lakeyev A, Rohn J, Kahl P (1997) Computational Complexity and Feasibility of Data Processing and Interval Computations Kluwer, Dordrecht
5. Kreinovich V, Longpré L, Starks SA, Xiang G, Beck J, Kandathi R, Nayak A, Ferson S, Hajagos J (2007) Interval versions of statistical techniques, with applications to environmental analysis, bioinformatics, and privacy in statistical databases. Journal of Computational and Applied Mathematics 199:418–423
6. Jaulin L, Kieffer M, Didrit O, Walter E (2001) Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics. Springer Verlag, London
7. Nguyen CH, Ho TB (2007) Kernel matrix evaluation. In: Manuela M. Veloso MM (ed) Proceedings of the 20th International Joint Conference on Artificial Intelligence IJCAI'07, Hyderabad, India, January 6-12, 2007, 987–992
8. Papadimitriou CH, Steiglitz K (1998) Combinatorial Optimization: Algorithms and Complexity, Dover Publications, Inc., Mineola, New York

9. Rabinovich SG (2005) Measurement Errors and Uncertainty. Theory and Practice, Springer Verlag, Berlin
10. Schölkopf B, Smola AJ (2002) Learning with Kernels, MIT Press, Cambridge, Massachusetts
11. Shawe-Taylor J, Cristianini N (2004) Kernel Methods for Pattern Analysis, Cambridge University Press, New York, New York
12. Vapnik VN (1995) The Nature of Statistical Learning Theory, Springer-Verlag, New York, New York
13. Wang L, and Chan KL (2002) Learning kernel parameters by using class separability measure. In: *NIPS's Sixth Kernel Machines Workshop*, Whistler, Canada, 2002.